- Joystick e paddle
- Informazione analogico-digitale
- Come controllare un Joystick - le interfacce
- STICK, STRING, PDL, PAD
- LEN, VAL, INSTR
- RIGHT$, MID$, LEFT$, SPACE$,
- Le operazioni sulle stringhe -slicing
- Videoesercizi
- Videogioco n° 9

VIDEOBASIC MSX
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore Editoriale: Roberto Pancaldi
Autore: Softidea -
Via Indipendenza 88-90 - Como
Redazione software:
Michele Casartelli
Francesco Franceschini
Progetto grafico:
Studio Nuovidea - via Longhi, 16 - Milano
Impaginazione:
Moreno Confalone
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

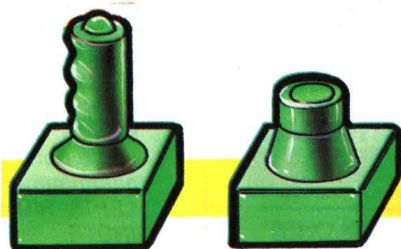
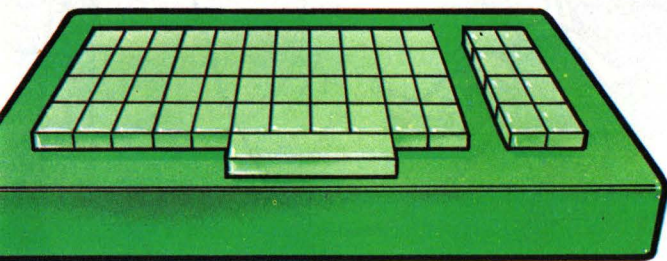
Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12
20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

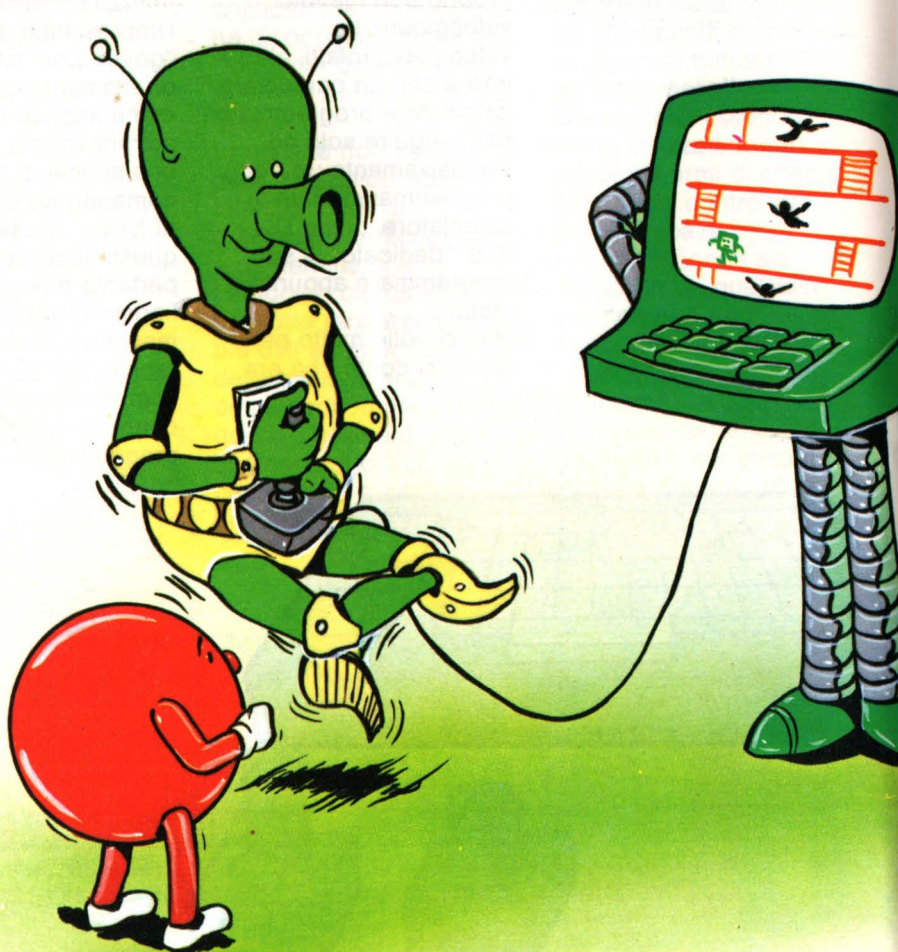
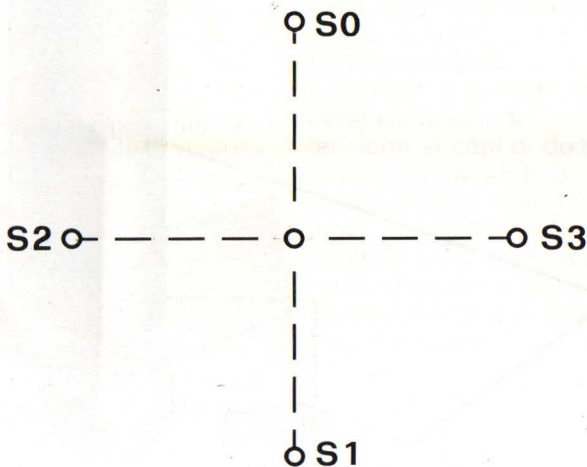
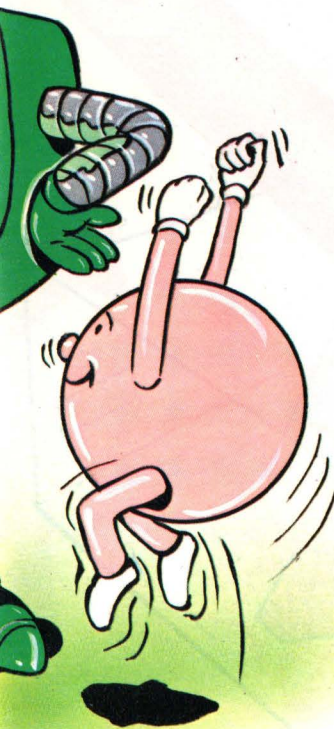
SOMMARIO

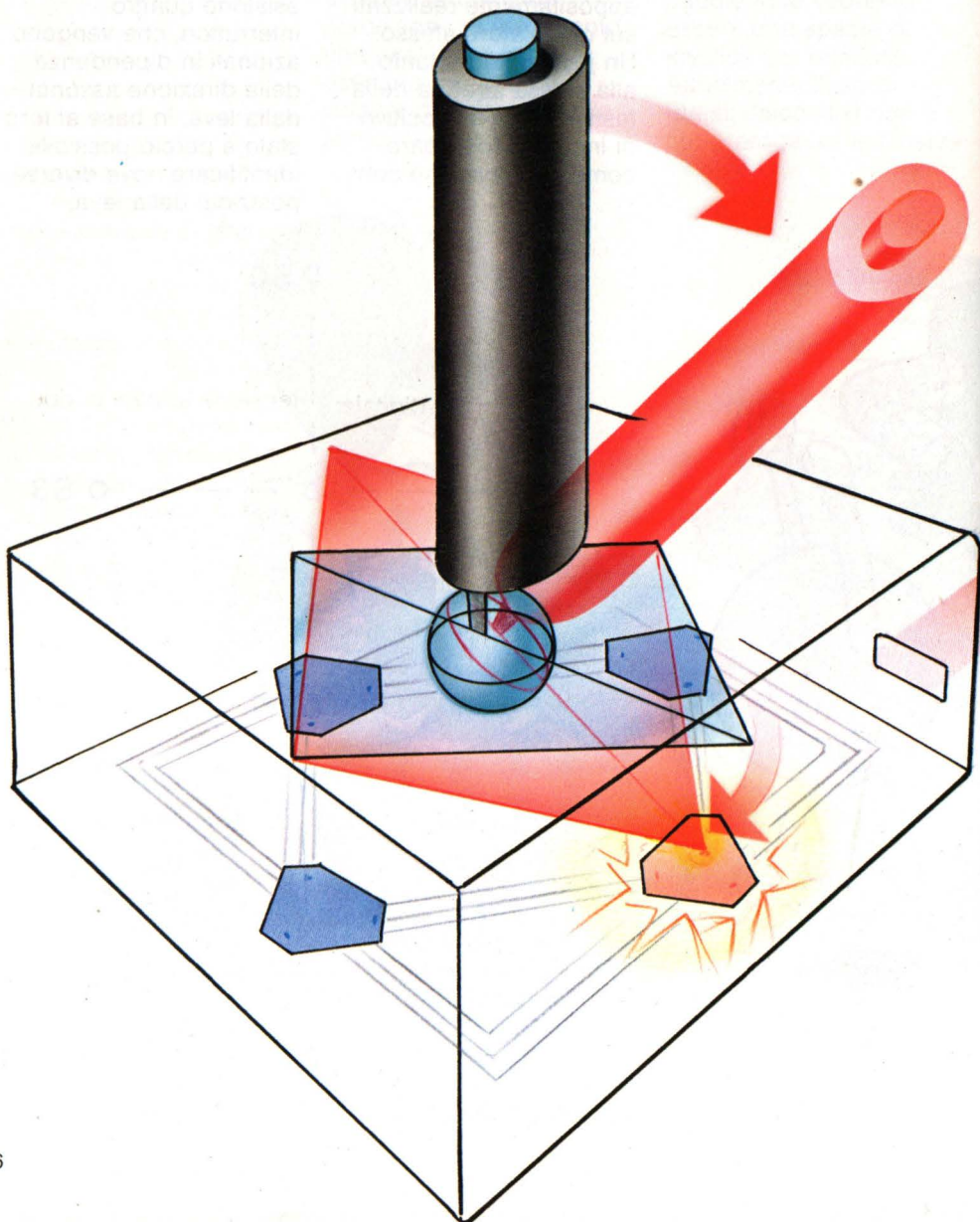
Introduzione

Chi non conosce il joystick, quella incandescente leva con pulsante onnipresente nei videogiochi da casa o da bar. Quante battaglie vinte o perse impugnandolo contro terribili nemici. Al di là del gioco, comunque il joystick (o le sorelle paddle) è una periferica di ingresso, con la sua logica (digitale) e i suoi principi di funzionamento, che è bene conoscere. Questo per quanto riguarda l'hardware.
Per la programmazione un altro prezioso tassello si aggiunge alle conoscenze già fatte: le funzioni di stringa.
Finora abbiamo limitato l'uso delle stringhe all'assegnazione, confronto e stampa. Vedremo in questa lezione come poter manipolare dei dati alfanumerici grazie a LEN, VAL, STR$, LEFT$, RIGHT$, MID$, SPACE$, INSTR.

Paddle e Joystick

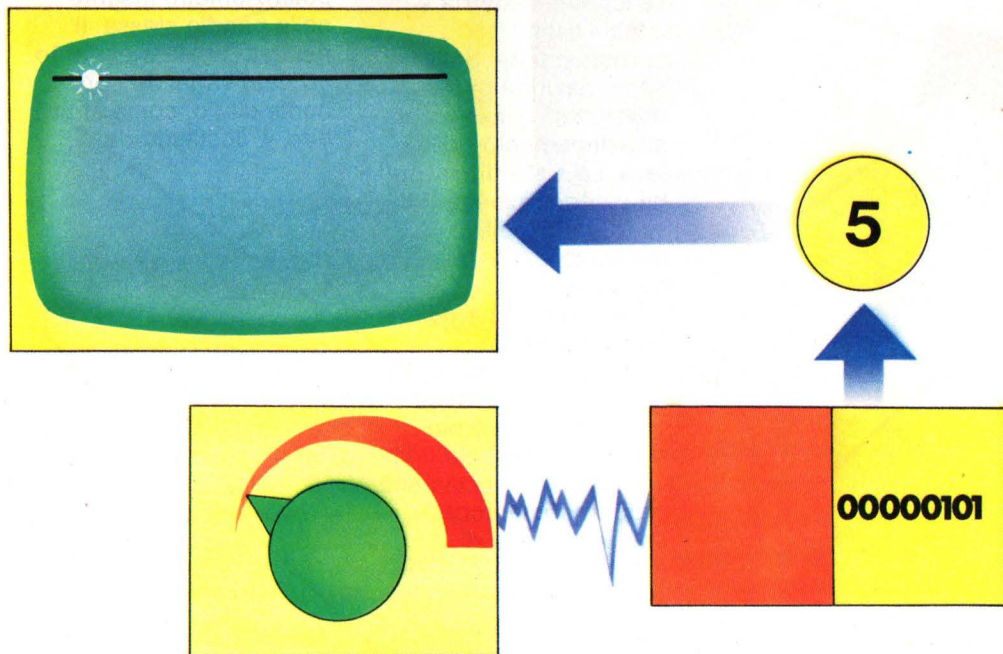
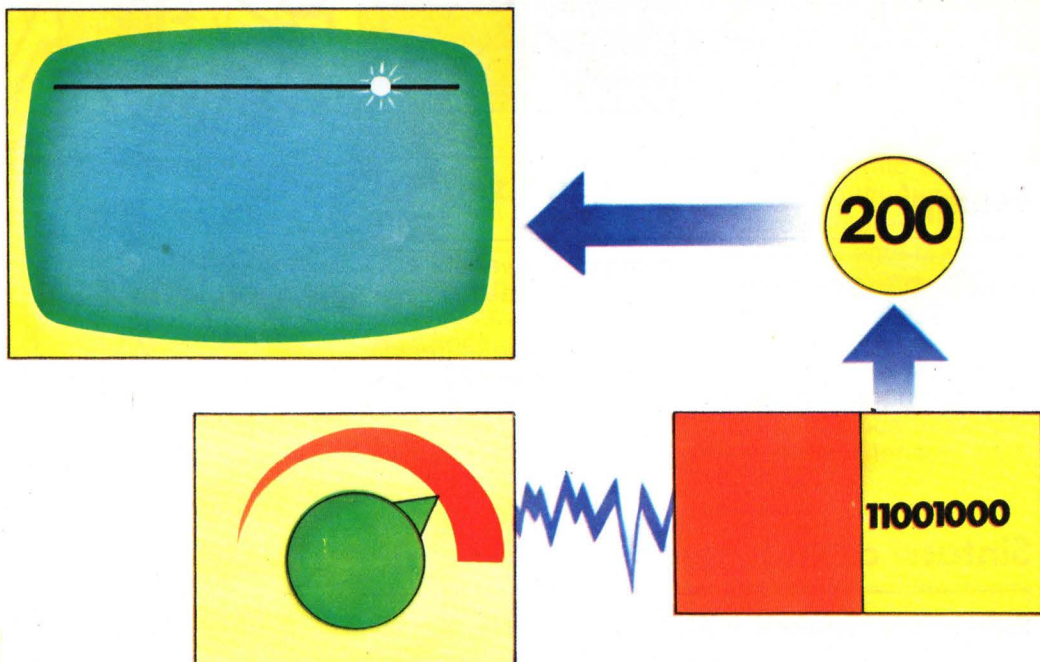
Abbiamo visto nelle scorse lezioni che il principale dispositivo di ingresso di dati ed informazioni per un calcolatore è costituito dalla tastiera, tant'è che praticamente non esistono computer - eccettuati quelli costruiti per particolari applicazioni - che non ne dispongano e che per l'input dei dati debbano ricorrere a mezzi differenti.
Principale non vuole però dire unico: sono infatti disponibili numerosi altri accessori e periferiche che in particolari circostanze possono rendersi molto più utili ed adeguati della tastiera, proprio come accade per l'uomo quando desidera qualcosa di più veloce o comodo della semplice (ma assolutamente indispensabile) penna per scrivere.
Oggi parleremo appunto di due di questi dispositivi: i joystick e le paddle.
Con ogni probabilità essi non ti sono completamente sconosciuti, dal momento che i loro nomi ricorrono molto spesso tra gli accaniti giocatori degli ormai diffusissimi ed onnipresenti videogiochi, essendo infatti utilizzati come principali armi di attacco e di difesa contro i vari invasori spaziali.
Anche chiunque sia entrato in un qualsiasi bar non può non conoscere un joystick, riconoscendo subito in tale nome l'incandescente levetta necessaria per evitare e sfuggire dai pericoli del "bombardamento

nemico".
Più facilmente non ti sarà invece molto noto il loro principio costruttivo e di funzionamento: l'obiettivo di questa lezione sarà appunto quello di spiegartene i segreti e le differenze.
Cerchiamo innanzitutto di vedere quali sono stati i motivi che hanno determinato la nascita e, soprattutto negli ultimi tempi, la diffusione dei joystick e delle paddle.
In pratica, tutto è cominciato pochi anni fa, proprio con i fatidici videogiochi: un videogioco, infatti, altro non è che un particolare calcolatore programmato per eseguire solo ed esclusivamente un certo programma; cioè un calcolatore, come si usa dire, "dedicato". Il suo programma è appunto il gioco.
Non ci volle molto per rendersi conto che era necessario superare la difficoltà della tastiera. In primo luogo la maggior parte dei tasti sarebbe stata assolutamente superflua per l'uso al quale era destinato il calcolatore. Inoltre sarebbe stato molto più semplice, e soprattutto divertente, per gli utilizzatori poter avere l'impressione di "governare" tutto direttamente, proprio come accade sulle astronavi vere, attraverso una specie di leva di comando ed un pulsante di fuoco. Attraverso questi accessori divenne pertanto possibile impartire ordini alla macchina, in modo rapido, facile e soprattutto senza conoscere nulla di programmazione.

Come controllare un joystick

Per una persona non è molto naturale premere dei pulsanti per comandare dei movimenti: non si avrebbe la continua percezione di movimento che ricaviamo per esempio ruotando il volante di un'automobile se dovessimo invece utilizzare dei tasti o dei pulsanti.
Fu così che nacque l'idea dei joystick e, conseguentemente, delle paddle (che vedrèmo essere una specie di joystick piu raffinati).
Attualmente l'usa di questi dispositivi non è più prerogativa esclusiva

dei videogiochi: quasi tutti i personal computer ne permettono difatti la connessione, attraverso alcuni collegamenti appositamente realizzati sul calcolatore stesso. Un joystick è pertanto - alla stessa stregua della tastiera - un dispositivo di input. Esso appare come una leva che con il suo movimento può permettere di spostare un oggetto sul video. Vediamo in quale modo. All'interno del joystick esistono quattro interruttori, che vengono azionati in dipendenza della direzione assunta dalla leva; in base al loro stato è perciò possibile identificare nove diverse posizioni della leva:

tutti aperti	leva verticale
S0 chiuso	nord
S1 chiuso	sud
S2 chiuso	ovest
S3 chiuso	est
S0, S1 chiusi	nord-ovest
S1, S2 chiusi	sud-ovest
S2, S3 chiusi	sud-est
S3, S0 chiusi	nord-est

A ciascuno di questi quattro interruttori viene associato, in una locazione della memoria all'interno del calcolatore, un bit: bit 1 per interruttore acceso o bit 0 per interruttore spento. Un quinto interruttore funge da

pulsante di fuoco o da segnalatore che la posizione desiderata è stata raggiunta.
Tramite programma basterà allora leggere nella locazione il valore di ciascuno di questi bit per risalire alla posizione della leva e quindi alla relativa azione da eseguire (spostamento del cursore, dell'astronave o dell'oggetto sul video).

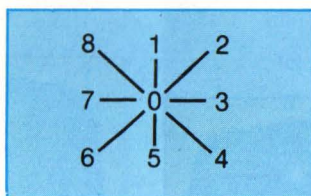
Informazioni analogiche e digitali: interfaccia

Il joystick è pertanto un tipico dispositivo digitale: il dato contenuto nella celletta di memoria è infatti l'esatto corrispondente della combinazione di interruttori conseguente all'azionamento della leva. Lo stato di ciascun bit dipende cioè dallo stato del rispettivo interruttore, senza nessuna approssimazione od arrotondamento.
Talvolta risulta però più comodo poter disporre di un dispositivo più sensibile a piccole variazioni e spostamenti, proprio come accade con un volante di automobile. Al joystick propriamente detto è stato quindi affiancato il cosiddetto joystick a potenziometro (paddle).
Ai fini del funzionamento una paddle è molto simile ad un joystick: tuttavia, anziché esservi una leva, nella paddle compare una manopola, la cui regolazione risulta pertanto molto più fine e precisa. La cosa viene inoltre ottenuta in modo completamente diverso da come accade con il joystick: lo spostamento della manopola provoca infatti non più l'azionamento di interruttori e pulsanti, ma delle variazioni di tensione ai capi di due resistenze variabili (o potenziometri) inserite nella paddle stessa. Il risultato finale è pertanto un dato variabile con continuità o, come si usa dire, è costituito da

un'informazione analogica. Un'opportuna conversione di tale informazione da analogica in digitale fornisce quindi, sotto forma di numero a otto bit, l'azione specificata dallo spostamento della manopola, memorizzandola in una particolare locazione della memoria. Per eseguire questa conversione viene naturalmente utilizzata un'apposita interfaccia, posta tra unità centrale e paddle, che ha lo scopo di trasformare il segnale continuo fornito dalla paddle in un'informazione binaria. Attraverso un'operazione di lettura, in questa locazione sarà allora possibile, analogamente a quanto succede col joystick, risalire al movimento desiderato.

Naturalmente, anche una paddle ha un suo limite di sensibilità, dovuto proprio a tale operazione di conversione; esso risulta comunque estremamente al di sopra di quello del joystick, permettendo in totale 256 possibili combinazioni. Non è però assolutamente detto che una paddle sia meglio di un joystick; dipende infatti dall'uso e dalla sensibilità necessaria alla singola applicazione. Ciò che sicuramente è più semplice è la connessione del joystick al computer, poiche non richiede alcun dispositivo di conversione, come invece è il caso della paddle.

STICK

Per rilevare la posizione dell'asta del joystick o lo stato dei tasti cursore, il Basic MSX mette a tua disposizione questa funzione.
Il parametro che è necessario fornire indica il joystick di cui ci vogliamo occupare; devi usare:

0 - per i tasti cursore
1 - per il joystick numero 1
2 - per il joystick numero 2

Il risultato restituito dalla funzione indica la direzione richiesta secondo questo codice:

Esempi

```
LET JO = STICK (0)
```

assegna alla variabile JO lo stato dei tasti cursore.
Se ad esempio la freccia a destra è premuta, JO conterrà il valore 3.

Sintassi della funzione

Espressione numerica = STICK (x)

La funzione STRIG

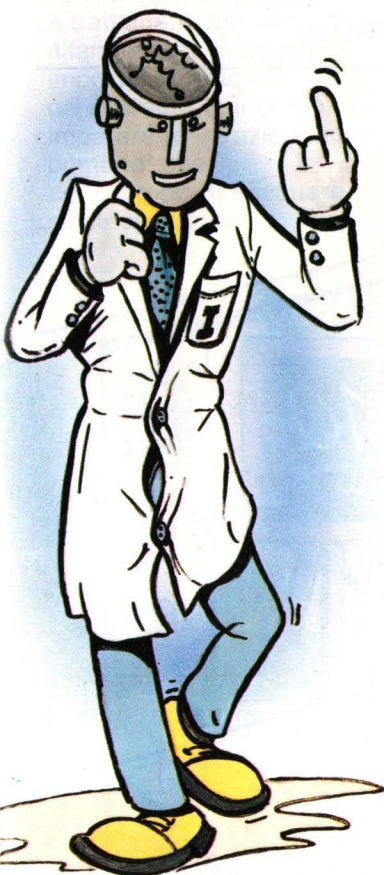
Questa funzione permette di conoscere lo stato dei pulsanti di fuoco dei joystick e della barra spaziatrice.
Il valore fornito da questa funzione può esere solo 0 o —1.

0 = tasto non premuto.
—1 = tasto premuto.

Il parametro della funzione STRIG serve ad indicare:

0 - barra spaziatrice
1 - pulsante 1 joystick 1
2 - pulsante 1 joystick 2
3 - pulsante 2 joystick 1
4 - pulsante 2 joystick 2

```
10 IF STICK (0) = 0
THEN 10
```

attende la pressione della barra spaziatrice.

Sintassi della funzione

Espressione numerica = STRIG (X)

La funzione PDL

Questa funzione permette di conoscere la posizione della manopola di una paddle.

Il valore fornito come risultato è un numero intero compreso tra 0 e 255, crescente man mano che la manopola viene girata verso destra (senso orario).
Il parametro è un numero compreso tra 1 e 12 ed indica la paddle a cui ci si riferisce.

Sintassi della funzione

Espressione numerica = PDL (x)

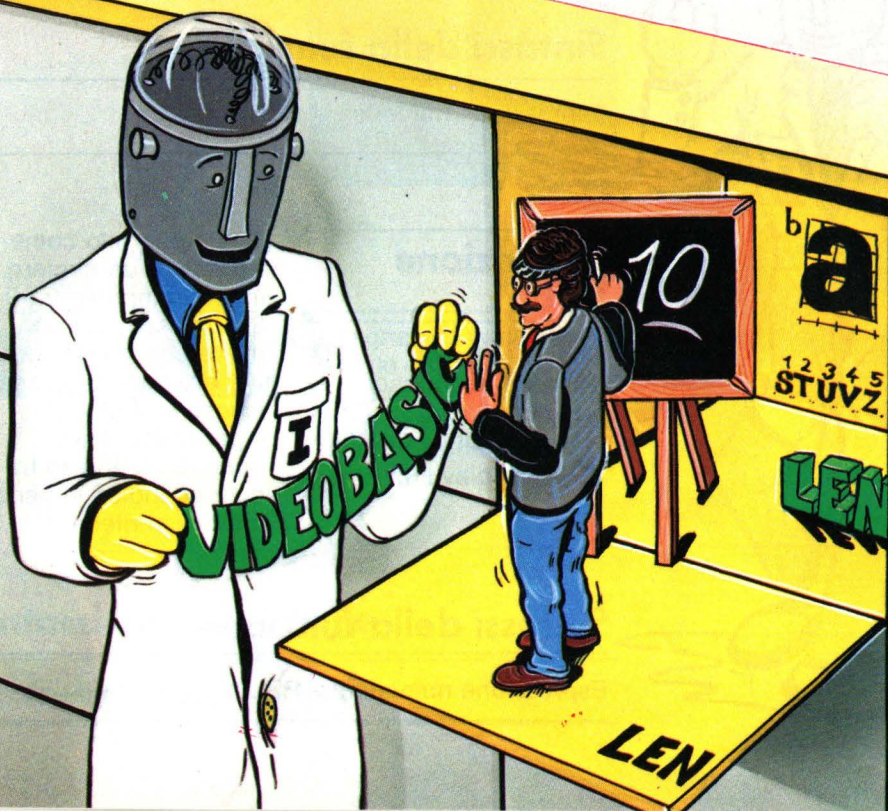
La funzione PAD

Questa funzione permette di conoscere lo stato di un touch pad. Ecco come funzione:

Pad (0) :—1 se toccato; 0 se no
Pad (1) :coordinata X (0-255)
Pad (2) :coordinata Y (0-255)
Pad (3) :—1 se pulsante premuto, 0 se no.

I parametri da 0 a 3 si riferiscono al touch pad inserito nella porta 1; quelli da 4 a 7 in modo analogo al touch pad inserito nella porta 2.

Sintassi della funzione

Espressione numerica = PAD (x)

Gli operatori di stringa

Fino a questo momento il nostro utilizzo delle stringhe si è limitato soltanto all'assegnazione, al confronto ed alla stampa delle espressioni di questo tipo. In molti programmi compaiono invece, in maniera estensiva, manipolazioni e trattamenti dei dati alfanumerici. In questa importante lezione cercheremo quindi di imparare le istruzioni e i comandi che consentono di effettuare simili operazioni.

LEN

La funzione LEN (abbreviazione dell'inglese LENght, lunghezza) è molto semplice da comprendere: consente infatti di trovare il numero di caratteri contenuti in una stringa. Per esempio, volendo determinare il numero di caratteri che compongono la stringa "GIOVANNI", sarà sufficiente impartire:

```
PRINT LEN ("GIOVANNI")
```

e sullo schermo apparirà 8.
La stringa "GIOVANNI", della quale vogliamo contare i caratteri, è l'argomento di LEN e va quindi messa entro parentesi, subito dopo la parola LEN.

LEN restituisce quindi un risultato numerico, partendo da un argomento rigorosamente di tipo stringa, variabile o costante. Da notare che le virgolette non vengono considerate (e d'altra parte è anche giusto è logico) come costituenti la stringa.
È un comando molto potente e di utilizzo assai frequente: i possibili casi nei quali può interessare conoscere la lunghezza di una stringa sono infatti numerosissimi, tra i quali - per esempio - gli incolonnamenti, le impaginazioni o i confronti tra stringhe.

Esempi

```
PRINT LEN ("CONTO CORRENTE")
```

Stampa il numero 14.

```
LET A = LEN ("")
```

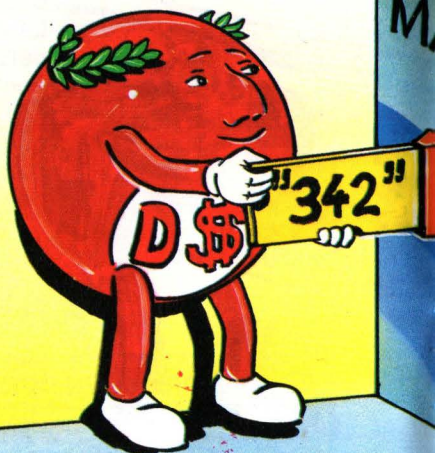
Alla variabile viene assegnato valore 0 (la stringa nulla non contiene alcun carattere).

```
LET A$ = "GIAN"
PRINT LEN (A$ + "CARLO")
```

```
LET A$ = "GIAN"
PRINT LEN (A$) + LEN ("CARLO")
```

Queste due istruzioni hanno lo stesso risultato, cioè 9. Nel primo caso viene prima eseguita la concatenazione delle due stringhe e quindi il calcolo della lunghezza, mentre nel secondo vengono separatamente calcolate le due lunghezze e quindi sommate.

Sintassi della funzione

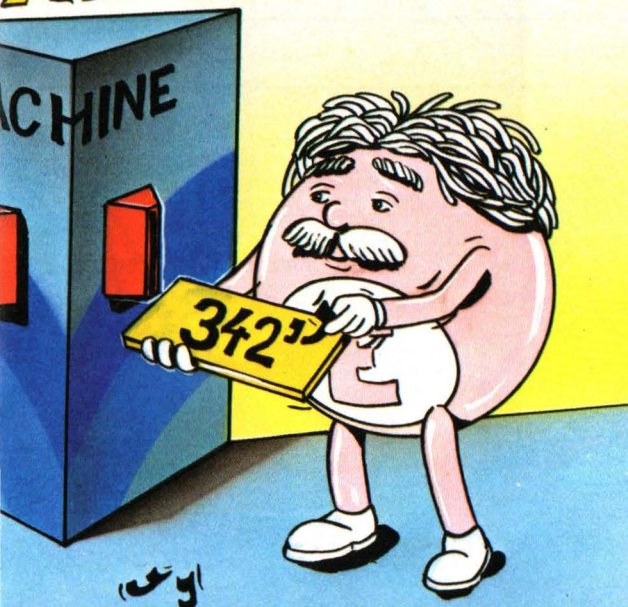
LEN (stringa)

VAL

Vi sono alcuni casi in cui risulta molto comodo poter convertire una stringa di numeri (come "3149") in un valore numerico, così da poterne far uso in espressioni numeriche. La funzione VAL (abbreviazione di VALue, valore) converte le stringhe in numeri. Così:

```
LET A = VAL ("3149")
```

assegna alla variabile numerica A il valore 3149.
Non bisogna comunque fraintendere questa funzione: essa non esegue infatti alcuna magia. Una stringa come "TAPPO" non potrà pertanto mai assumere alcun valore numerico; quando l'interprete BASIC incontra un'istruzione come

```
PRINT VAL ("TAPPO")
```

convenzionalmente assegna allora al risultato valore 0, per quanto la cosa possa sembrare non aver molto senso.

Esempi

LET X = VAL ("8")	Assegna alla variabile X il valore numerico 8.
PRINT VAL ("54C3")	Stampa il numero 54, arrestando cioè la conversione al primo carattere non numerico.
LET P$ = "103" LET Q$ = "2" PRINT P$ + Q$	Stampa la somma (cioè la concatenazione) delle stringhe P$ e Q$. Quindi il risultato è la stringa "1032".
PRINT VAL (P$) + VAL (Q$)	Stampa la somma dei valori numerici 103 e 2, cioè 105.

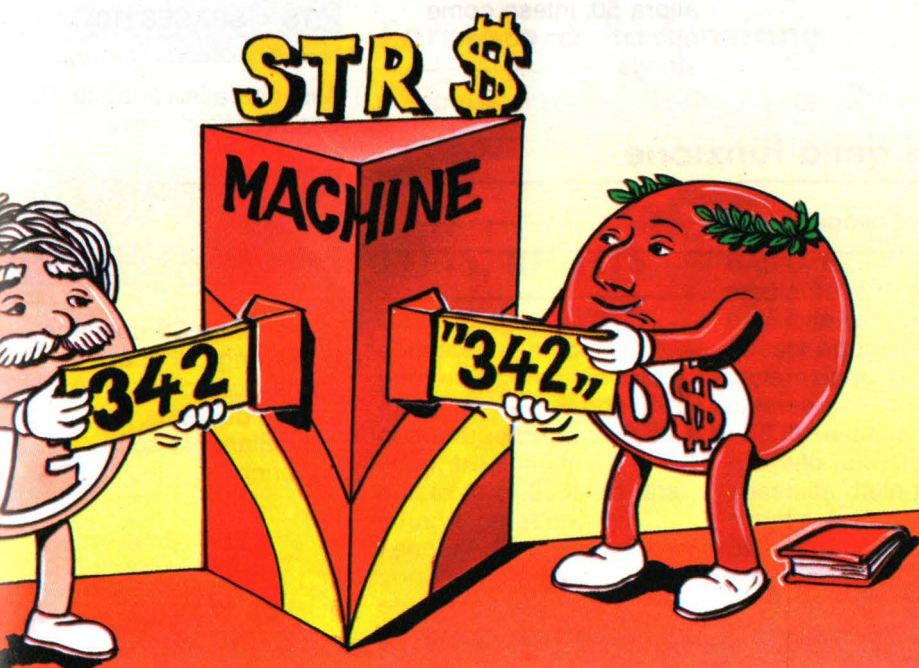
Sintassi della funzione

VAL (stringa)

STR$

STR$ (abbreviazione di STRing, stringa) è la funzione reciproca di VAL: essa consente infatti di convertire una quantità numerica in una stringa di caratteri.
Quando desideriamo trasformare un numero (come 8) in una stringa, in modo che possa essere combinato con altre stringhe (per esempio "mila"), STR$ è la funzione che fa al caso nostro.
L'istruzione

```
PRINT STR$ (8) + " " + "MILA"
```

produrrà allora come risultato la stampa della stringa "8 MILA".

Esempi

LET A$ = STR$ (50)	Assegna ad A$ il valore alfanumerico "50".
LET S$= STR$ (3 * 7– 2)	Risultato: S$ = "19".
PRINT VAL (STR$ (50))	È un'istruzione inutile, per quanto perfettamente lecita. Il valore numerico 50 viene infatti prima convertito in stringa (con STR$) e quindi ritrasformato in numero. Sullo schermo comparirà allora 50, inteso come numero, non come stringa.

Sintassi della funzione

STR$ (espressione)

SPACE$

Questa funzione fornisce una stringa lunga a piacere e costituita da caratteri " " (spazio vuoto).
Il parametro della funzione indica la lunghezza della stringa che si desidera ottenere; naturalmente non deve essere maggiore di 255 pena l'errore "Illegal function call".

```
T$ = SPACE$ (10)
```

Assegna alla variabile T$ una stringa di spazi.

```
N$ = "GIORNATA
PIOVOSA"
C$ = SPACE$ (LEN(N$))
```

La variabile C$ contiene una stringa di spazi lunga quanto N$. Può essere utilizzata per cancellare N$ sullo schermo.

Sintassi della funzione

SPACE$ (espressione)

INSTR

La funzione INSTR opera su due stringhe con lo scopo di ricercarne la seconda all'interno della prima, partendo da carattere più a sinistra. Se la trova, restituisce un numero indicante dal quale carattere della prima stringa inizia quella ricercata; in caso contrario restituisce 0.

```
PRINT INSTR
("VIDEOBASIC","BASIC")
```

stampa 6 perchè la stringa "BASIC" è contenuta nella stringa "VIDEOBASIC" a partire dal sesto carattere.
Alla funzione INSTR è inoltre possibile fornire anche un parametro numerico per specificare da quale carattere della prima stringa iniziare la ricerca.
Questo parametro va messo prima delle stringhe e separato da una virgola.

```
PRINT INSTR
(3, "VIDEOBASIC", "I")
```

stampa 9 e non 2 perchè la ricerca è iniziata dal terzo carattere.

Formato della funzione

<espress. num.> = INSTR ([N,] stringa, sottostringa)

LEFT$, RIGHT$

L'operazione di concatenamento permette di aggiungere caratteri (ricordandosi comunque di non superare il limite massimo di 255!) ad una stringa, caratteri, eseguendo cioè una sorta di somma tra stringhe.
In certi casi, anziché aggiungere, potrebbe essere necessario dover estrarre dei caratteri da una stringa: esistono allora nel linguaggio BASIC delle funzioni mediante le quali è possibile eseguire questa operazione di separazione in maniere diverse.
LEFT$ restituisce come risultato una stringa costituita dalla parte sinistra (left) della stringa utilizzata come argomento, per il numero di caratteri specificato.

Quindi:

```
PRINT LEFT$ ("GIANCARLO",4)
```

stampa i primi quattro caratteri della stringa "GIANCARLO", partendo dalla sinistra, cioè GIAN. RIGHT$ ha le stesse regole di LEFT$, ma esegue l'operazione dalla parte opposta: essa consente infatti di separare un certo numero di caratteri dalla stringa di partenza, cominciando questa volta dalla parte destra (right).
Allora

```
PRINT RIGHT$ ("GIANCARLO",5)
```

fornirà come risultato la visualizzazione sul video di "CARLO".
Nota come sia LEFT$ che RIGHT$ chiedano due argomenti: uno di tipo stringa ed uno di tipo numerico. Il primo specifica la stringa sulla quale si dovrà operare l'estrazione dei caratteri, il secondo indica invece il numero dei caratteri interessati dall'estrazione stessa.

Esempi

```
LET A$ = "gennaiofebbraiomarzo"
PRINT LEFT$ (A$, 7)
```

Stampa "gennaio"

```
LET A$ = "gennaiofebbraiomarzo"
PRINT RIGHT$ (A$, 5)
```

Stampa "marzo"

```
PRINT LEFT$ (("NASTRO"), LEN ("NASTRO") – 1)
```

Il numero di caratteri da prelevare partendo da sinistra sarà 5 (infatti LEN ("NASTRO") dà 6, che, diminuito di 1, risulta poi 5).
Sullo schermo apparirà allora "NASTR".

```
PRINT RIGHT$ (("NASTRO"), LEN ("NASTRO") – 1)
```

Tutto è uguale all'esempio appena visto, tranne che si parte da destra anziché da sinistra. Il risultato sarà allora la visualizzazione di "ASTRO".

```
PRINT RIGHT$ ("GIORNO", 7)
```

Poiché il valore 7 è maggiore della lunghezza totale della stringa, verrà stampata la stringa completa, cioè "GIORNO".

```
LET A$ = LEFT$ (A$, – 1)
```

Errore! Non è possibile estrarre – 1 caratteri da una stringa!
(Illegal function call).

Il seguente, breve programma (altri ne troverai sulla parte della lezione dedicata alla programmazione) ti illustra una semplice applicazione dei concetti finora esposti:

```
10 INPUT "SCRIVI UNA PAROLA QUALSIASI"; P$
20 FOR I = 1 TO LEN (P$)
30 PRINT LEFT$ (P$, I); "   "; RIGHT$ (P$, I)
40 NEXT I
50 END
```

Sintassi delle funzioni

LEFT$ (stringa, numero caratteri)

RIGHT$ (stringa, numero caratteri)

MID$

MID$ permette l'estrazione di un certo numero di caratteri, partendo da un qualsiasi punto della stringa originaria. È sicuramente l'istruzione più potente del gruppo di comandi per la suddivisione delle stringhe (può infatti facilmente sostituire sia LEFT$ che RIGHT$). MID$ si serve di tre argomenti: la stringa di origine, il numero che indica da quale carattere si deve iniziare l'estrazione ed infine il numero di caratteri da estrarre.

```
PRINT ("GRUPPO EDITORIALE JACKSON", 8, 10)
```

stampa quindi sullo schermo la stringa "EDITORIALE", prelevata dalla stringa di partenza, cominciando dalla ottava posizione e contando 10 caratteri.
L'utilizzo classico di questo comando lo si ha in quei programmi che richiedono in ingresso una data scritta nella forma GG/MM/AA (come 27/10/60). Con MID$ è allora molto facile scomporre la data nelle sue tre componenti GG, MM e AA.
Vediamo subito come fare:

```
10 INPUT "SCRIVI LA DATA (GG/MM/AA)"; D$
20 LET G$ = MID$ (D$, 1, 2)
30 LET M$ = MID$ (D$, 4, 2)
40 LET A$ = MID$ (D$, 7, 2)
```

Alla fine di questo programma G$, M$ e A$ conterranno rispettivamente il giorno, il mese e l'anno della data inizialmente assegnata a D$.

"PORCOSPINO"

Esempi

LET A$ = "gennaiofebbraiomarzo" PRINT MID$ (A$, 8, 8)	Stampa "febbraio" Nota che con MID$ si possono ottenere anche gli stessi risultati delle funzioni LEFT$ e RIGHT$.
LET A$ = "gennaiofebbraiomarzo" PRINT MID$ (A$, 1, 7)	Stampa "gennaio"
LET A$ = "gennaiofebbraiomarzo" PRINT MID$ (A$, 16, 5)	Stampa "marzo"
LET K$ = MID$ ("CUORE", 1, 8)	Assegna alla variabile K$ l'intera lunghezza di "CUORE" - partendo dalla prima posizione - visto che il numero di caratteri è inferiore agli 8 desiderati.
PRINT MID$ ("VIDEOBASIC", – 1, 3)	Dal momento che la posizione di un carattere in una stringa non può essere inferiore a 1, anziché un risultato, apparirà quindi sullo schermo un messaggio di errore.

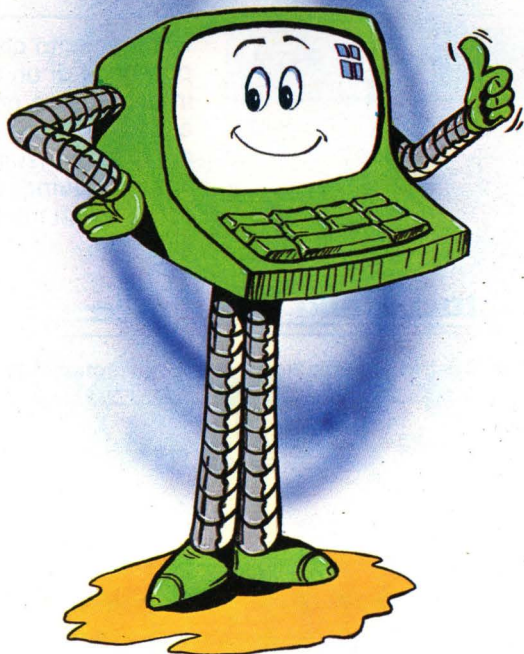
Sintassi della funzione

MID$ (stringa, posizione iniziale, numero caratteri)

Operazioni sulle stringhe

I programmi di oggi ti illustreranno alcune possibili applicazioni sulle funzioni per il trattamento delle stringhe.

Vediamo il primo esempio: stampare una stringa, introdotta dalla tastiera, solo se il primo carattere è diverso da "A" o da "B".

```
 10 CLS: PRINT "INSERISCI LA STRINGA";
 20 INPUT N$
 30 REM A$ DIVENTA LA PRIMA LETTERA DELLA STRINGA
 40 LET A$ = LEFT$ (N$, 1)
 50 REM CONTROLLA SE INIZIA PER A O B
 60 IF (A$ = "A") OR (A$ = "B") THEN GOTO 100
 70 REM LA STRINGA INIZIA CON UNA LETTERA DIVERSA
 80 PRINT "LA STRINGA È : "; N$
 90 END
100 PRINT "LA STRINGA NON È STAMPABILE PERCHÉ INIZIA PER "; A$
```

Particolari difficoltà non ne esistono; l'unica cosa da notare è la riga 60: è infatti necessario controllare che il primo carattere sia diverso da "A" o da "B".

Secondo esempio: scrivere una parola - come al solito introdotta dalla tastiera - in senso inverso, cioè opposto al normale (da destra verso sinistra).

```
10 CLS: PRINT "INSERISCI LA STRINGA";
20 INPUT A$
30 LET K = LEN (A$)
40 FOR I = K TO 1 STEP - 1
50 PRINT MID$ (A$, I, 1);
60 NEXT I : PRINT
70 INPUT "ANCORA"; R$
80 IF R$ = "S" OR R$ = "s" THEN RUN
90 END
```

Anche in questo caso il procedimento è molto semplice: si preleva un carattere alla volta dalla parola (o dalla frase) da scrivere, partendo dalla destra e per tante volte quanti sono complessivamente i caratteri costituenti la parola stessa. Alla fine sullo schermo si troverà visualizzata la stringa originaria scritta al rovescio.
Se qualcosa non ti è chiaro, aggiungendo la riga

```
55 FOR J = 0 TO 500
   : NEXT J
```

introducendo cioè un ciclo di ritardo, avrai la possibilità di osservare molto più agevolmente la formazione, carattere dopo carattere, della parola completa.

```
 10 CLS: PRINT "INSERISCI LA STRINGA";
 20 INPUT A$
 30 LET B$ = ""
 40 LET K = LEN (A$)
 50 FOR I = 1 TO K
 60 LET A$ = LEFT$ (A$, K - I + 1)
 70 LET C$ = RIGHT$ (A$, 1)
 80 LET B$ = B$ + C$
 90 NEXT I
100 PRINT B$
110 END
```

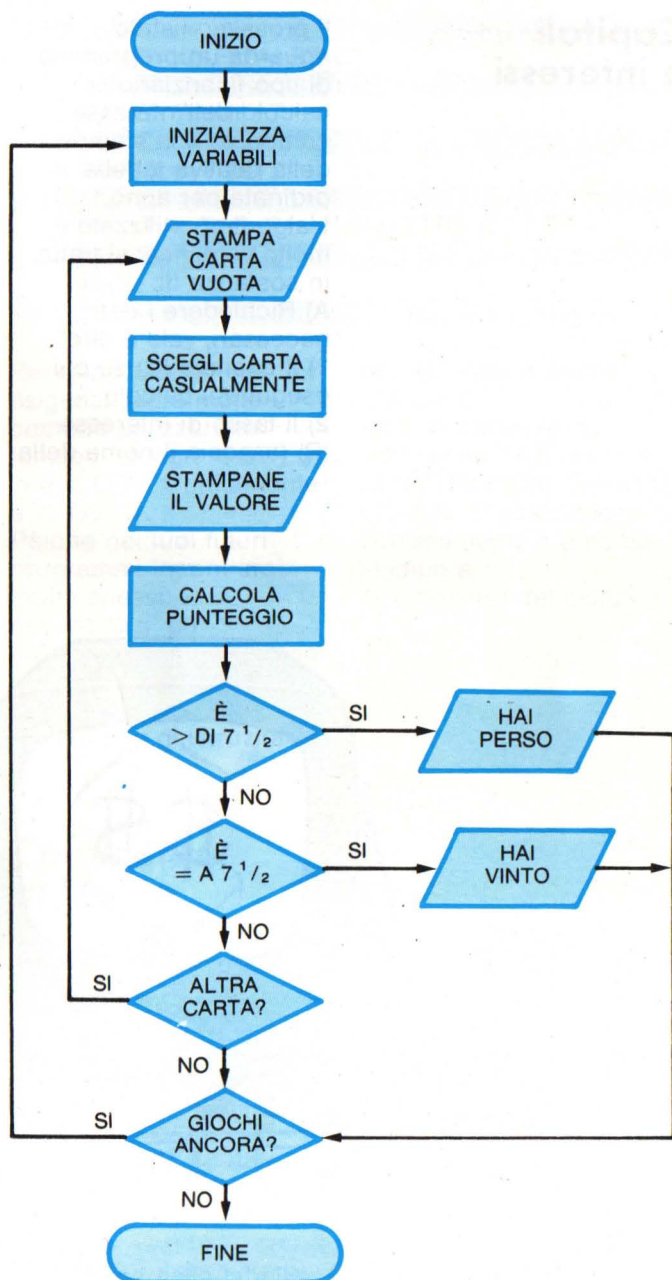
In questo caso i vari caratteri, anziché essere direttamente inviati sullo schermo, vanno ad aggiungersi, uno dopo l'altro, nella variabile B$. Terminato il ciclo (linee 30 - 70) B$ conterrà la parola invertita e la si potrà quindi stampare.

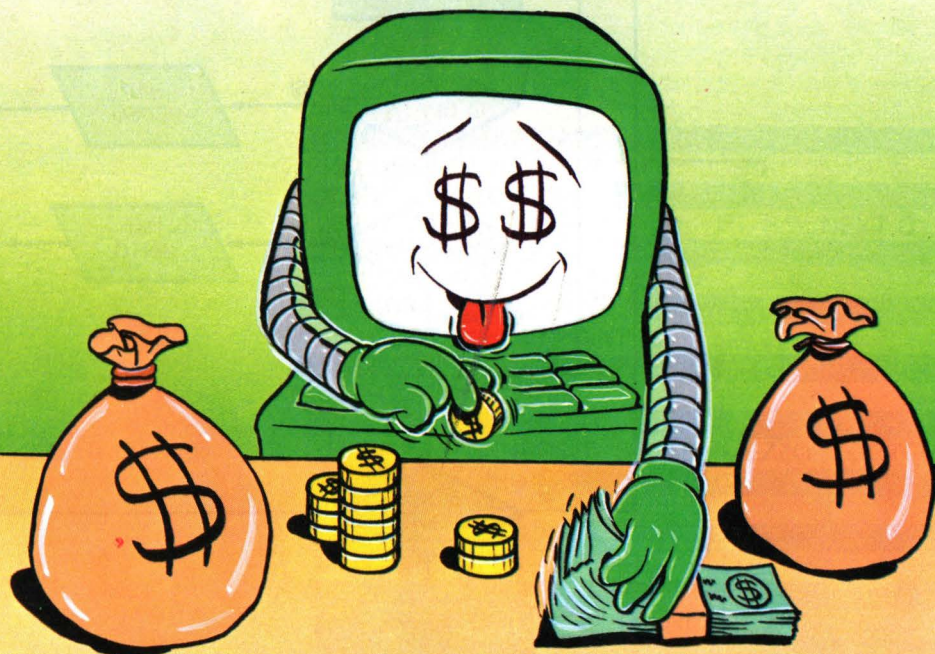
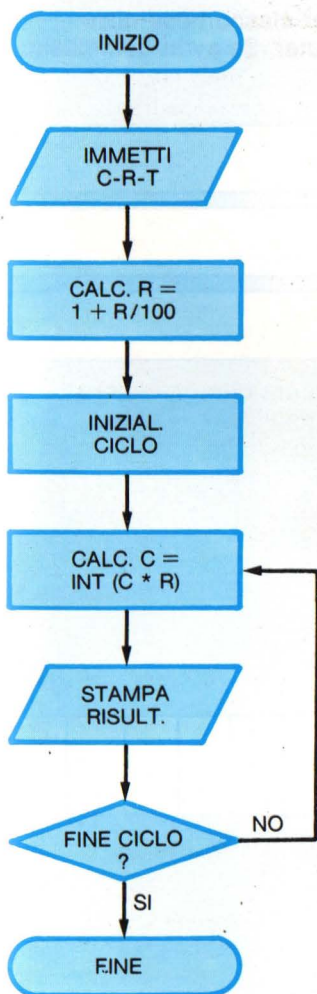
Settemmezzo

Le regole di questo gioco sono molto semplici: le carte dall'asso (1) al 7 valgono tanti punti quanti ne rappresentano, mentre le "figure" soltanto 1/2 punto.
Si possono chiedere quante carte si vuole e vince chi ottiene esattamente 7 punti e 1/2. Attenzione, però, a non "sballare", a non superare, cioè, il fatidico punteggio.
Nel programma, il tuo MSX ha il compito di gestire il gioco: si occupa, infatti, della distribuzione delle carte, del controllo scrupoloso del punteggio raggiunto e dell'invio dei messaggi appropriati.
Il cuore del problema sta nella simulazione dell'uscita delle carte. Questo compito è svolto dalle righe 10, 35, 40 del listato, mentre alla 15, 20, 25, 30, 45 e 50 è affidato quello di visualizzarle.
Tutto è basato sulla stringa "A234567JQK" assegnata ad A$ e sulle funzioni MID$ e RND per realizzare una corretta estrazione casuale.
Le righe dalla 55 alla 70 controllano il punteggio

```
10 A$="A234567JQR" :P=0 : CLS
15 LOCATEN, 0 : PRINT " ┌─────────┐ "
20 FORC=1 TO7
25 LOCATEN, C : PRINT " │         │ " : NEXT C
30 LOCATEN, 8 : PRINT " └─────────┘ "
35 C=INT (RND(—TIME)*10)+1
40 B$=MID$(A$,C,1)
45 LOCATEN+1,1 : PRINTB$
50 LOCATEN+4,7 : PRINTB$ : PRINT : PRINT
55 IFC>7THENC=.5
60 P=P+C
65 IFP=7.5THENPRINT "SETTEMMEZZO" : GOTO90
70 IFP>7.5THENPRINT "HAI PERSO" : GOTO90
75 INPUT "CARTA" ; R$
80 REM
85 N=N+3 : IFR$="S"ORR$="s" THEN15
90 INPUT"UN'ALTRA PARTITA" ; R$
95 IFR$="S"ORR$= "s" THENRUN
100 END
```

e stampano i messaggi opportuni, mentre la 75, 80, 85, consentono di richiedere un'altra carta. L'ultima parte (righe 90 e 95) permette di giocare una nuova partita.

Il contorno delle carte è stampato con i caratteri grafici ottenibili coi tasti GRPH e R, —, Y, V, N, SHIFT\.

Capitali e interessi

Il prossimo listato riguarda un programma di tipo finanziario: il calcolo dell'interesse composto e la stampa della relativa tabella ordinata per anno.
L'algoritmo utilizzato è molto semplice: si tratta, in sostanza di:
A) Richiedere i dati necessari, vale a dire:
1) Il capitale (C) su cui eseguire il calcolo.
2) Il tasso di interesse (R) (associa il nome della variabile al termine < RAGIONE >)
3) Il tempo (T).
B) Calcolare il montante (capitale + interesse) per ogni singolo anno, avendo cura di considerare come capitale per l'anno successivo il montante del precedente.
C) Stampare il prospetto dei vari anni, avendo cura che gli importi risultino bene incolonnati.

```
 10 CLS
 20 INPUT "CAPITALE"; C
 30 INPUT "INTERESSE"; R
 40 INPUT "ANNI"; T
 50 LET R = 1 + R/100
 60 FOR A = 1 TO T
 70 LET C = INT (C * R)
 80 PRINT A; TAB (21 – LEN (STR$ (C))); C
 90 NEXT A
100 END
```

Per quest'ultima esigenza, sarà molto comodo utilizzare le funzioni stringa < STR $ > e < LEN > unitamente a < TAB >.
Poichè nei tuoi futuri programmi incontrerai molto spesso la necessità di un incolonnamento rigoroso dei dati, è bene che familiarizzi ora con questo tipo di istruzioni.

Commento al listato

20-40 input dei valori di CAPITALE, TASSO (R) e TEMPO (N. ANNI)
50 Trasformazione del tasso di interesse da percentuale a decimale
60 Impostazione del ciclo in funzione degli anni
70 Calcolo del montante annuo
80 Stampa della tabella con N. anno e valore montante.
Nota in particolare l'istruzione TAB (21 – LEN (STR$ (C))) che ha il compito di posizionare correttamente il dato da stampare.
90 Chiusura del ciclo.

VIDEOESERCIZI

Annota nello spazio apposito il risultato da te previsto per ciascun esercizio proposto e poi verificarlo con la soluzione del tuo computer. Se avrai commesso anche un solo errore, ripassa la lezione.

```
10 LET A$ = "12345"
20 LET B$ = "67"
30 LET C$ = "890"
40 PRINT LEN (BS)
50 PRINT LEN (A$) + B$ + C$)
60 PRINT LEN (C$) — LEN (B$)
```

Per evidenziare

```
10 T$ = "TITOLO"
20 FOR C = 1 TO LEN (T$)
30 PRINT MID$ (T$, C, 1); " ";
40 NEXT C
```

Prima di far girare il programma scrivere l'output

Indovina dove stampa le stringhe

```
10 CLS : WIDTH32
20 INPUT "stringa =": S$
30 LOCATE (16—LEN (S$)/2), 10 : PRINTS$
40 PRINT : PRINT "ancora?"
50 R$ = INKEY$ : IFR$ = " " THEN50
60 IFR$ = "s" ORR$ = "S" THENRUN
70 END
```

In quale riga?

Dove rispetto alle colonne?